International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

A Multimodal Approach for Detecting AI Generated Content using BERT and CNN

Vismay Vora, Jenil Savla, Deevya Mehta, Dr. Aruna Gawade
Department of Computer Engineering
Dwarkadas J. Sanghvi College of Engineering
Mumbai, India
voravismay9@gmail.com, jenilsavla20@gmail.com, deevyamehta02@gmail.com, aruna.gawade@djsce.ac.in

Abstract—With the advent of Generative AI technologies like LLMs and image generators, there will be an unprecedented rise in synthetic information which requires detection. While deepfake content can be identified by considering biological cues, this article proposes a technique for the detection of AI generated text using vocabulary, syntactic, semantic and stylistic features of the input data and detecting AI generated images through the use of a CNN model. The performance of these models is also evaluated and benchmarked with other comparative models. The ML Olympiad Competition dataset from Kaggle is used in a BERT Model for text detection and the CNN model is trained on the CIFAKE dataset to detect AI generated images. It can be concluded that in the upcoming era, AI generated content will be omnipresent and no single model will truly be able to detect all AI generated content especially when these technologies are getting better.

Keywords-Synthetic Data; LLMs; Generative AI; BERT; CNN; Multimodal Detection.

I. INTRODUCTION

Everyone has been a witness of this transition into the era of technology. Smart Machines and Artificial Intelligence are the latest technology being developed. Intelligent systems like ChatGPT and BingAI have been in recent talks. These LLM-based chatbots generate tons of content daily. Users haven't wasted much time integrating the use cases of these models into their tasks. The provision to get a curated response just on the input of a single prompt has given new powers to humans. Generating content has become really easy now. A couple of years back the development of any content required the human effort of scanning through various resources, obtaining responses from different places and then combining them to formulate the final data. The traditional barriers to obtaining such content have lowered. Now, generated content has become easily accessible to everyone.

This new virtue has even introduced its own limitations. The need for responsible generation and use of content is very important. Users now generate information for anything and make that information available on the internet. But the credibility of such content is still questionable. Some open discussion forums like StackOverflow had to temporarily ban users from posting the answers generated by AI. The reason given was that the forum might get flooded with such answers which initially may seem right but later be proved wrong on careful examination. These AI models can be trained by the user and then provide responses based on that. So individuals can run propaganda by using it to spread false information which is actually inputted by them.

Students use this privilege to write essays and do similar tasks which require cognitive thinking, killing the objective of the task. Thus, there is a need to keep a check on AI generated content. The present paper discusses how to identify the generated text, images and deepfake content so that there is a filtered and curated generation and distribution of such content.

II. RELATED WORK

The first modality is deepfake videos which is a synthetic approach that uses deep learning algorithms to replace the person in an existing image or video with someone else's characteristic or likeness. It can be used for either lawful or malevolent reasons, including entertainment, education, propaganda, misinformation, harassment, blackmail, and so on. A systematic literature review (SLR) conducted by [1] summarised 112 relevant papers from 2018 to 2020 that provided a variety of techniques. They classified them into four categories: deep learning-based approaches, traditional machine learning-based methods, statistical techniques, and blockchain-based techniques. The paper [2] explores new methods for detecting deepfake videos, specifically focusing on faces using advanced generative models like VAEs and GANs. The authors combine various types of Vision Transformers with a convolutional EfficientNet B0 used as a feature extractor, obtaining comparable results with some very recent methods that use Vision Transformers. They present a straightforward inference procedure based on a simple voting scheme for handling multiple faces in the same video shot. The best model achieved an AUC of 0.951 and an F1 score of 88.0%, very close to the state-of-the-art on the Deep-Fake Detection Challenge (DFDC) [3]. Lastly, [4] proposes an approach to

***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

detect facial manipulation in video sequences using an ensemble of different trained Convolutional Neural Network (CNN) models. The proposed solution combines different models obtained from a base network (EfficientNetB4) using attention layers and siamese training. The approach is tested on two publicly available datasets with over 119,000 videos and shows promising results in detecting deepfakes and other facial manipulation techniques such as Face2Face, FaceSwap, and NeuralTextures. Furthermore, an explainable deepfake detection framework is discussed in [5] as well as a literature review covering the aspects of generation, detection and applications is present in [6].

One of the largely distributed forms of generated content is text. The Large Language Models are improving every day. Lets's look at some literature about detecting AI generated text. Article [7] is an article published in MIT technology review which discusses various methods to detect AI generated text. It mentions the use of Large Language Models themselves by retraining them on the human written text so as to be able to distinguish it. The use of watermarks during generation is also suggested which will easily separate the AI generated text. This article concludes that the best way to spot the generated text is by use of human intellect as the content presented might not be in a way a generic reader prefers. The research work [8] displays the direct use of generative AI models to detect the content generated by AI. But this will only be accurate for the model that both the systems are trained on.

A differential analysis of AI generated content is presented in [9]. The focus of this article is evaluating scientific content generation. A feature description framework is developed to identify the subtler errors in AI generated text by examining the syntax, semantics and pragmatics. The writing style was considered as a distinction parameter. The authors summarise a number of model and distribution independent functions for detection tasks in various domains and the insights in this paper help guide the optimization of AI models to produce high-quality content and address related ethical and security concerns.

The traceability of AI generated text is discussed in [10]. This article highlights the techniques commonly used to make AI generated text undetectable. It mentions how the use of a paraphrasing tool can make text easily pass various detectors. The detection model is compared to a random classifier. The authors also present a solution to bypass the use of watermarks. This is done via the means of spoofing attacks where generated text is modified to contain the hidden signature and be detected as human generated text. The research article [11] provides details on the use of watermarks for separating the two kinds of texts. These watermarks are short spans of text within the entire text which are invisible to humans. They can only be detected by the use of specific algorithms. The authors also test their technique using a model from the Open Pretrained Transformer (OPT) family where the use of watermarks is concluded to be robust and secure. The paper [12] extensively compared various implementations of BERT in natural language processing, specifically focusing on medical narrative analysis.

Moving onto the third modality which is images it was found that [13] presents a systematic study on the detection of CNN-generated images by exploiting the systematic shortcoming present in these images in replicating high-frequency Fourier spectrum decay attributes. However, study concludes that high-frequency Fourier spectrum decay discrepancies are not inherent in existing CNN-based generative models. In a recent study, article 14] proposed an optimization-assisted autoregressive method that significantly improves image demosaicing using deep convolutional neural networks.

Another recent study generates a synthetic dataset that mimics the CIFAR-10 dataset [15], including complex visual attributes like photorealistic reflections and provides the CIFAKE dataset [16]. This dataset sets up a binary classification problem to differentiate between real and AI-generated images. It proposes a minimal neural network architecture to perform this binary classification and detect fake and real images. Moreover, the study employs explainable AI techniques to identify relevant features for classification, revealing that small visual imperfections in the background play a crucial role.

Despite these efforts, the area of research for the detection of AI generated content still has a lot of scope for improvement such as explaining the decisions made by the model during detection. The authors of [17] conducted a detailed review outlining the application of Explainable AI (XAI) in pain modeling using machine learning techniques. The current research mainly focuses on the detection of AI generated content in the text and image modalities. This paper proposes a feature-based model training approach for text detection and a neural network-dependent classification method for images. These techniques are discussed in detail in the following section.

III. METHODOLOGY

The proposed AI generated content detection techniques for both modalities are explained in two different sections. Detection of the text generated by the Large Language Models (LLMs) using feature driven classification model is presented in sub-section A whereas the AI generated image identification method using a multi-layered convolutional neural network is elaborated in the sub-section B.

A. AI Generated Text Detection

Detection of AI generated text is a fairly tedious task. This generated text detection process can be subdivided into 3 major steps which include data preprocessing, feature extraction step and the model selection & training step. These steps themselves comprise of tasks that are further explained in detail.

1) Data Preprocessing: In current article, the dataset used is from a kaggle contest - ML Olympiad Detect ChatGPT Answers [18]. The training and test sets contain prompts, which are questions on various things and of multiple categories, and the replies to those prompts in the form of grammatically correct paragraphs. The prompts include opinion-based questions, general knowledge questions, scientific facts, and special ones to induce complexity. Each prompt contains an unannounced ratio of human to ChatGPT generated replies, per example ChatGPT generated reply, 5 human generated replies). This dataset is converted into representational form by applying data cleaning. The unlabelled data points and rows with null values are removed. The data is then split into an 80:20 ratio as training and validation sets respectively. The model will be trained on the training set whereas the validation test will be used to test the model and generate evaluation metrics.

2) Feature Extraction: The feature extraction step requires a labeled dataset. This dataset is curated to only included representational and complete data. For feature extraction, one or more of the proposed features can be used as per the specific characteristics of the prepared dataset. The following are the features that can be selected –

- *Vocabulary based features:* There are two useful vocabulary based features - vocabulary size and lexical richness. The vocabulary size measures the diversity and richness of word usage. The AI model is expected to present a specific vocabulary pattern that contains rare words and technical terms. Lexical Richness means a measure of Token-Type Ratio (TTR) which is the ratio of unique words to the total number of words. The lexical richness of the AI generated will be less than that of human-generated text due to the limited training data. Thus, vocabulary based features can be used to distinguish the two.
- *Syntactic features:* Every text has some syntactic features which vary from writer to writer. The distribution of Parts of Speech (POS) tags such as nouns, verbs, adjectives or adverbs can be analyzed. The AI generated text can exhibit different syntactic patterns. AI generated text might show simpler sentence construction. So, the syntactic complexity of sentences can be measured using metrics like average sentence length, the number of clauses or syntactic depth tree.
- *Semantic features:* AI generated text might lack semantic coherence which results in abrupt topic shifts or incoherent transitions between sentences and paragraphs. Techniques like word embeddings can be used to compare the similarity between sentences. AI generated text might exhibit lower semantic similarity compared to human generated text due to a lack of contextual understanding.
- *Stylish features:* The stylistic features of text such as sentence structure, tone, formality and context can also be analyzed. Also, the stylish variations can be measured using features such as sentence length variability and the use of punctuation marks. When compared to human generated text, AI generated text might have different stylish patterns and exhibit inconsistencies in writing style.

For the feature extraction step, the vocabulary based, syntactic, semantic and stylish features are considered. The vocabulary feature extraction is done by first tokenizing the text, then determining the vocabulary size by considering only the words which are alphabetic and aren’t any stopwords. This is then used for calculating the lexical richness of the text which is the ratio of the number of unique words to the total words in the text. Syntactic features chosen are count of parts of speech (POS) tags such as nouns and verbs. The nltk library’s SemanticIntensityAnalyzer is used to extract the semantic scores of each sentence in the text. Finally, average sentence length and punctuation count are used as stylish features for the text.

3) Model Selection and Training: The next step is of model selection and training. This involves the selection of appropriate machine learning or deep learning model for detection of AI generated text. This model can be anything including Logistic Regression, Support Vector Machines (SVM), Random Forests, Gradient boosting models (XGBoost, LightGM), Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and their variants (LSTM, GRU, etc) or Transformer based models (BERT, GPT, etc). This model is selected on the specific characteristics and requirements of the detection task such as size of dataset, feature representation, interpretability and computational resources available. For training the model, the dataset is first processed into appropriate representation using techniques such as bag-of-words, TF-IDF, word embeddings (eg Word2Vec, GloVe) or contextual embeddings (eg BERT, GPT). The model is be then evaluated using metrics such as accuracy, loss and other relevant measures. Based on the validation set's performance, the model's hyperparameters can then be fine-tuned.

The model selected for this research is BERT (Bidirectional Encoder Representations for Transformers), which is one of the most effective models for the detection AI generated text. This model is pre-trained on a large amount of data which means it has learned rich representations of words and sentences, capturing complex semantic and syntactic relationships. BERT also generates contextualized word embeddings which enable a better understanding of the content. Owing to all these factors, BERT was selected as the model for this experiment.

For training this model, the dataset is first loaded from a CSV file, split into training and testing sets, and then tokenized using BERT tokenizer. The BERT model is initialized, and the tokenizer's encodings are used to create TensorFlow Dataset objects for training and testing. The BERT model is fine-tuned on the training dataset, and the model's performance is evaluated on the test dataset using accuracy as the evaluation metric.

4) *Experimental setup:*

- *Optimization Strategy*: Adam optimizer was employed as the optimization strategy. Adam combines the advantages of AdaGrad and RMSProp and is well suited for training deep neural networks. It dynamically adjusts the learning rate based on the gradients, resulting in effective weight updates
- *Loss Function and Evaluation Metrics*: The Sparse Categorical cross-entropy loss function, which is commonly used for multi-class classification was used. The logits argument is set to True as normalized probabilities aren't used. This loss function compares these logits with the integer target labels and computes the loss based on the categorical cross-entropy formula. The evaluation metrics employed are accuracy, precision, and recall. Accuracy provides an overall assessment of the model's predictions, while precision measures the proportion of true positive predictions out of all positive predictions, and recall measures the fraction of true positive predictions out of all actual positive instances.

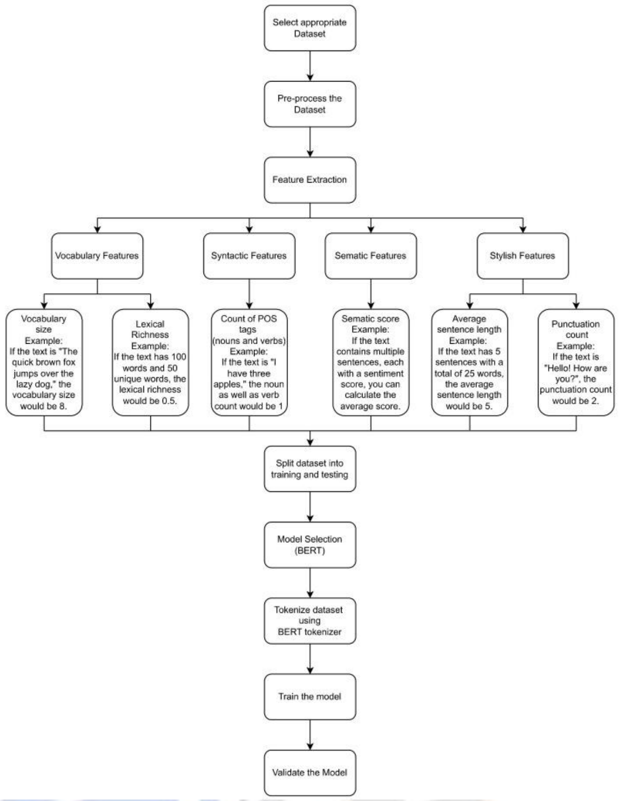
Select appropriate Dataset
Pre-process the Dataset
Feature Extraction
Vocabulary Features
Syntactic Features
Sematic Features
Stylish Features
Vocabulary size Example: If the text is "The quick brown fox jumps over the lazy dog," the vocabulary size would be 8.
Lexical Richness Example: If the text has 100 words and 50 unique words, the lexical richness would be 0.5.
Count of POS tags (nouns and verbs) Example: If the text is "I have three apples," the noun as well as verb count would be 1
Sematic score Example: If the text contains multiple sentences, each with a sentiment score, you can calculate the average score.
Average sentence length Example: If the text has 5 sentences with a total of 25 words, the average sentence length would be 5.
Punctuation count Example: If the text is "Hello! How are you?", the punctuation count would be 2.
Split dataset into training and testing
Model Selection (BERT)
Tokenize dataset using BERT tokenizer
Train the model
Validate the Model

Figure 1. Steps in invloved in AI Generated Text Detection

- *Training Process*: The model is trained for a fixed number of epochs. Each epoch represents a complete pass through the training dataset. In this study, the model is trained for 3 epochs. The batch size is set to 16, meaning that a set of 16 input strings is processed before updating the model's weights. This approach ensures faster training and evaluation of the model.

B. *AI Generated Image Detection*

In the previous subsection the method for detecting AI generated text using BERT model is discussed in detail, this subsection explains the technique for detection of AI generated images by implementing a multi-layer Convolutional Neural Network (CNN) architecture.

1) *Data Preprocessing:*

The CIFAKE dataset [16] consists of 60,000 synthetically created images and 60,000 real photos obtained from Krizhevsky and Hinton's CIFAR-10 dataset [17]. The dataset is divided into two classes: REAL and FAKE. The REAL images were extracted from the CIFAR-10 dataset. To construct the equivalent of CIFAR-10 for the FAKE pictures, Stable Diffusion version 1.4 is used. The dataset contains 100,000 images for training (50,000 images per class) and 20,000 images for testing (10,000 images per class).

- *Image Resizing:* This is the first step in data preprocessing which was done while loading the training data. The image_size option was set to (32, 32), which resized the images to 32x32 pixels.

- *Normalization:* After resizing the images, normalization was performed by dividing the pixel values by 255.0 and converting them to floating-point values in the range [0,1]. This step ensures consistent scaling of the input data and facilitates faster convergence during model training.

2) Model Architecture:

The model architecture to detect AI generated images is a convolutional neural network which consists of the following components:

- *Input Layer:*
  The input shape is defined based on the size of the images in the CIFAKE dataset, and a sequential layer is created using the tf.keras.Sequential command.
- *Rescaling Layer:*
  A rescaling layer is added to normalize the pixel values of the input images by 255.0, ensuring consistent scaling of the input data.
- *Convolutional Layers:*
  The model starts with a Conv2D layer with 32 filters, a 3x3 kernel size, and the ReLU activation function. This layer extracts features from the input images using convolutional techniques. A MaxPooling2D layer is applied after the first convolutional layer to downsample the feature maps and retain the most significant features. Another Conv2D layer with 64 filters and a 3x3 kernel size is added, also using the ReLU activation function.

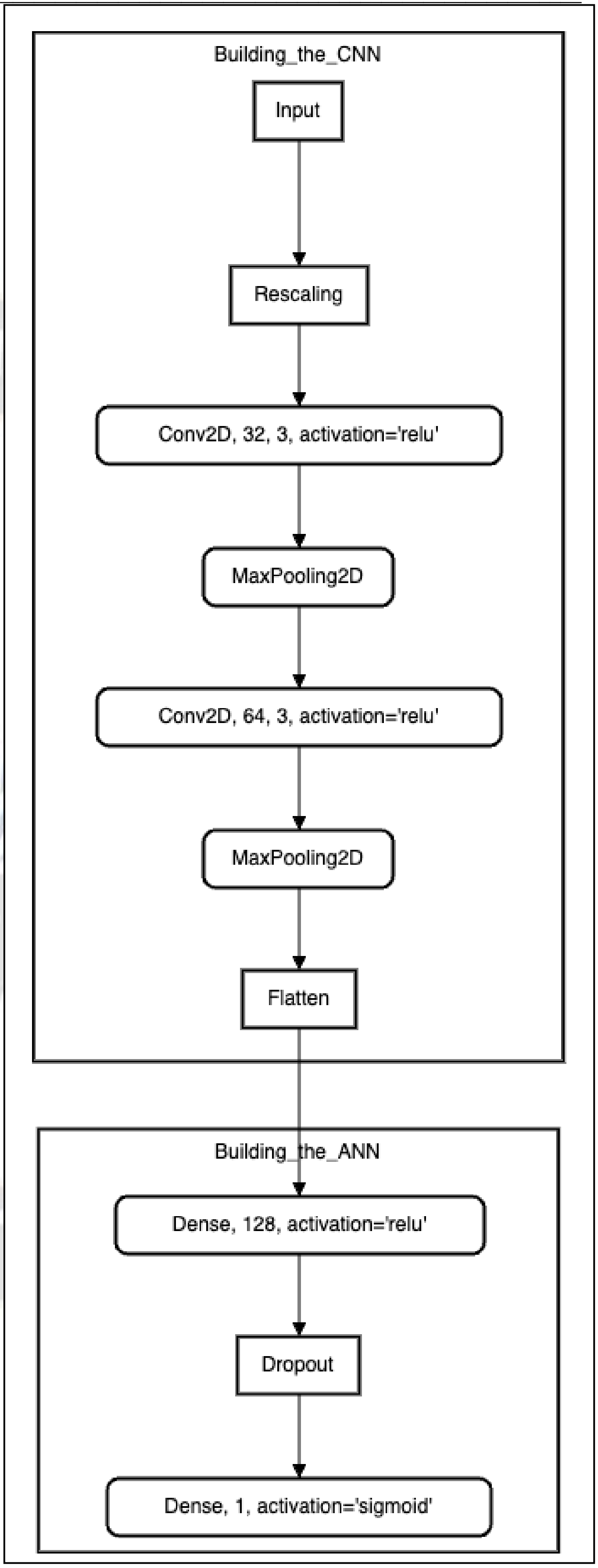
Building_the_CNN
Input
Rescaling
Conv2D, 32, 3, activation='relu'
MaxPooling2D
Conv2D, 64, 3, activation='relu'
MaxPooling2D
Flatten
Building_the_ANN
Dense, 128, activation='relu'
Dropout
Dense, 1, activation='sigmoid'

Figure 2. Architecture of CNN Model for AI Generated Image Detection

Convolution involves sliding a small window (kernel/filter) over the input image and computing dot products between the kernel and the corresponding pixels in the window. MaxPooling2D improves the model's translation invariance and generalization ability by downsampling the feature maps.

- *Flattening Layer:*
After the convolutional layers, a flattening layer is applied to convert the 2D feature maps into a 1D feature vector. This prepares the data for the fully connected layers of the artificial neural network (ANN).
- *Fully Connected Layers (ANN):*
The flattened feature vector is then processed by a Dense layer with 128 units and the ReLU activation function. This layer captures high-level abstract representations of the feature attributes. To mitigate overfitting, a Dropout layer with a dropout rate of 0.5 is added, randomly disabling 50% of the neurons during training. Finally, an output layer with a Dense layer having 1 unit and the sigmoid activation function is included. This layer generates the prediction probabilities for the binary classification task, indicating whether an image is generated by AI or not.
- *Model Compilation:*
The model is compiled using the Adam optimizer, binary cross-entropy loss function, and evaluation metrics such as accuracy, precision, and recall.

3) Experimental Setup:

- *Optimization Strategy:*
Adam optimizer was employed as the optimization strategy. Adam combines the advantages of AdaGrad and RMSProp and is well-suited for training deep neural networks. It dynamically adjusts the learning rate based on the gradients, resulting in effective weight updates.
- *Loss Function and Evaluation Metrics:*
The binary cross-entropy loss function, which is commonly used for binary classification tasks was used. It minimizes the gap between predicted and true probabilities. The evaluation metrics employed are accuracy, precision, and recall. Accuracy provides an overall assessment of the model's predictions, while precision measures the proportion of true positive predictions out of all positive predictions, and recall measures the fraction of true positive predictions out of all actual positive instances.
- *Training Process:*
The model is trained for a fixed number of epochs. Each epoch represents a complete pass through the training dataset. In this study, the model is trained for 10 epochs. The batch size is set to 32, meaning that a set of 32 image samples is processed before updating the model's weights. This approach ensures faster training and evaluation of the model.

IV. RESULTS AND DISCUSSION

The results observed after implementation of two above mentioned techniques are discussed in this section. It has two subsections where the results for text detection model are mentioned in subsection A and the visualizations for the image detection model are presented in section B.

A. Text Detection

The AI generated text detection was performed using a BERT Classifier model which was trained over 3 epochs With batch size of 16. After every epoch the loss function and accuracy was calculated. Observed value of loss after first epoch was 0.5012 yielding an accuracy of 0.7701, which was bettered in the next epoch reducing loss to 0.3194 with an accuracy of 90.44%. After training of final epoch, the loss identified was 0.2576 and accuracy in this epoch was 90.10%.

This trained model was then evaluated over the validation dataset. Similarly, SVM and Random Forest Classification models were also trained and evaluated on the same dataset. Following are the results obtained:

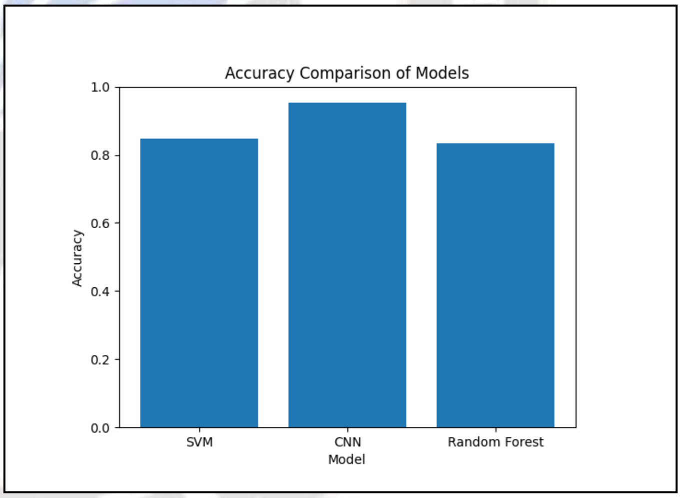

Figure 3. Accuracy Comparison Plot

The accuracy comparison graph (Fig. 3) shows a box plot of accuracy achieved by each model. All these models were evaluated over same data to maintain consistency. The BERT model showed highest accuracy of 0.8999, followed by Random Forest Classifier with accuracy 0.7867 and SVM with accuracy 0.7334.

The Confusion Matrices in Fig. 4, Fig. 5 and Fig. 6 for BERT, SVM and Random Forest Classifier respectively provide insights about distribution of true positive, true negative, false positive, and false negative predictions. Again, it was observed that BERT achieved a higher number of true positives and true negatives compared to SVM and Random Forest. However, it

Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

exhibited a slightly higher number of false positives and false negatives as well, which is an area of improvement.

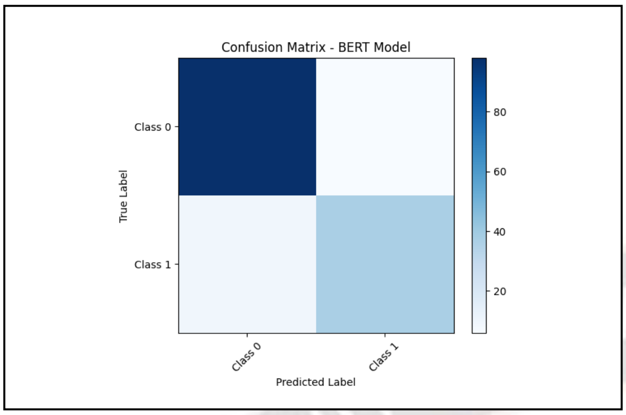

Figure 4. Confusion Matrix (BERT Model)

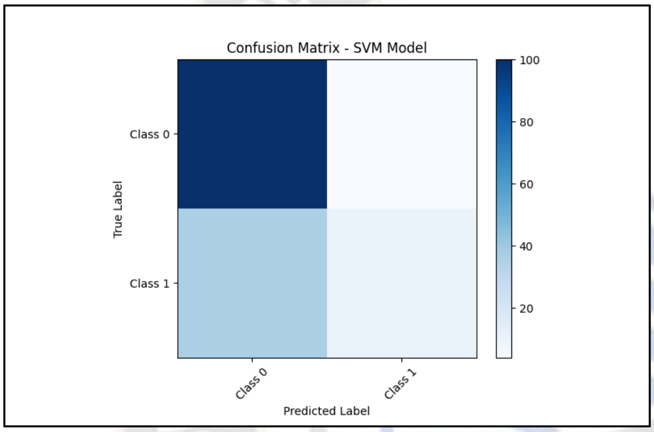

Figure 5. Confusion Matrix (SVM Model)

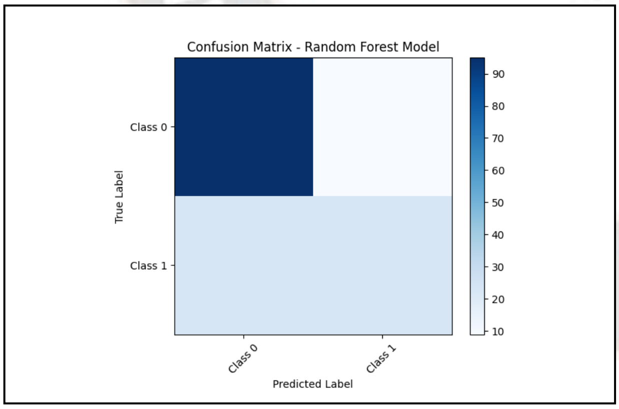

Figure 6. Confusion Matrix (Random Forest Model)

The ROC curves (Fig. 7) illustrate the ability of the models to distinguish between positive and negative classes. It can be observed that the ROC curve is inclined towards the top left of the graph. BERT achieved the highest area under the curve (AUC) value of 0.90, indicating its strong discriminatory power. SVM attained an AUC of 0.79, while Random Forest achieved an AUC of 0.82. The higher AUC values indicates the better performance of BERT and SVM in correctly classifying the text.

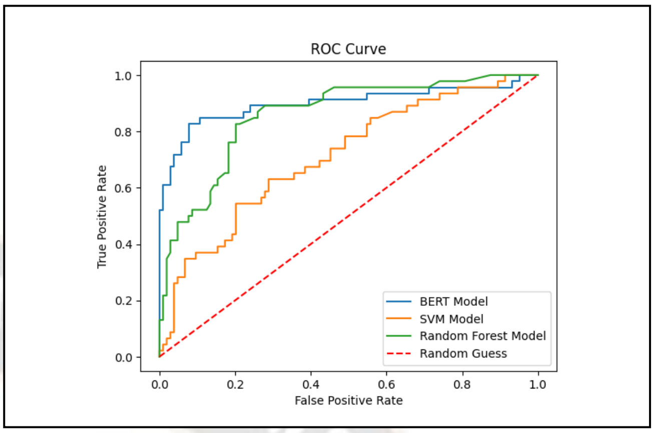

Figure 7. ROC Curve

Below graph is a F1 score curve (Fig. 8) depict the balance between precision and recall for each model. The F1 score for BERT was the highest, indicating its ability to achieve a balance between precision and recall. SVM and Random Forest exhibited slightly lower F1 scores, suggesting potential trade-offs between precision and recall in their predictions.

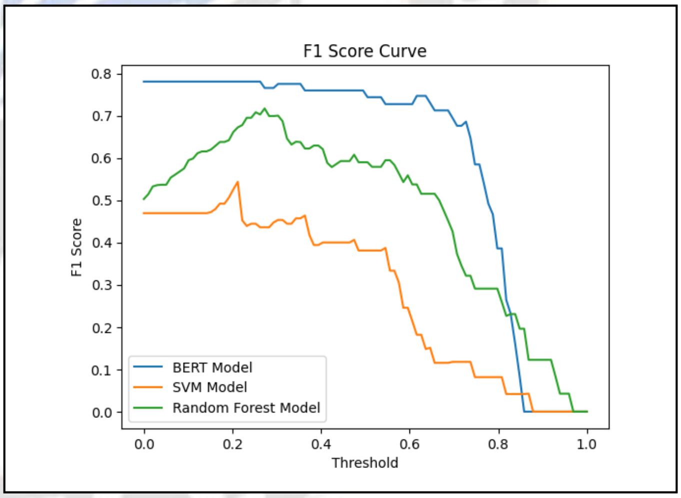

Figure 8. F1 Score Plot

The area under the precision recall curve (Fig. 9) is very high for BERT model which determines high precision and high recall, followed by Random Forest Classifier and then SVM. High values for precision and recall indicate that the model will accurately predict the classes in majority of cases.

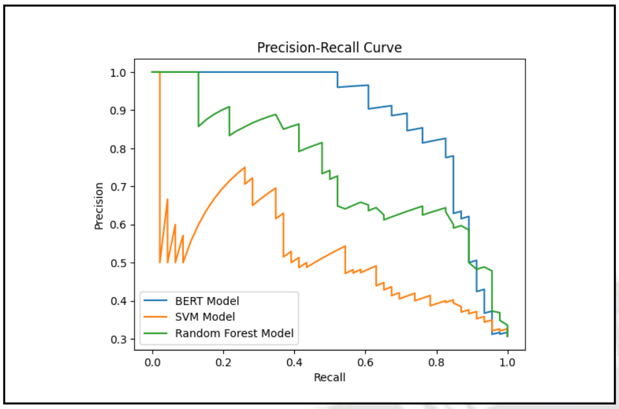

Figure 9. Precision Recall Plot

After evaluating all the above metrics, it can be observed that the BERT Classification model outperforms both SVM and Random Forest Classifier in detecting the AI generated text using the selected features for classification task.

B. *Image Detection*

The results for the CNN model used to detect AI generated images are as follows:

The CNN-ANN model was trained and evaluated on the CIFAKE dataset to detect AI-generated images. The model produced promising findings, demonstrating its ability to discern between AI-generated and real-world images. The model acquired a low loss value of 0.1929, indicating its ability to minimise the error between predicted and real labels after training for 10 epochs. On the validation dataset, the model's accuracy was 93.55%, indicating a high overall correct prediction rate. The model training metrics namely the accuracy and loss for both the training and validation steps are shown as separate plots in Fig. 10 above.

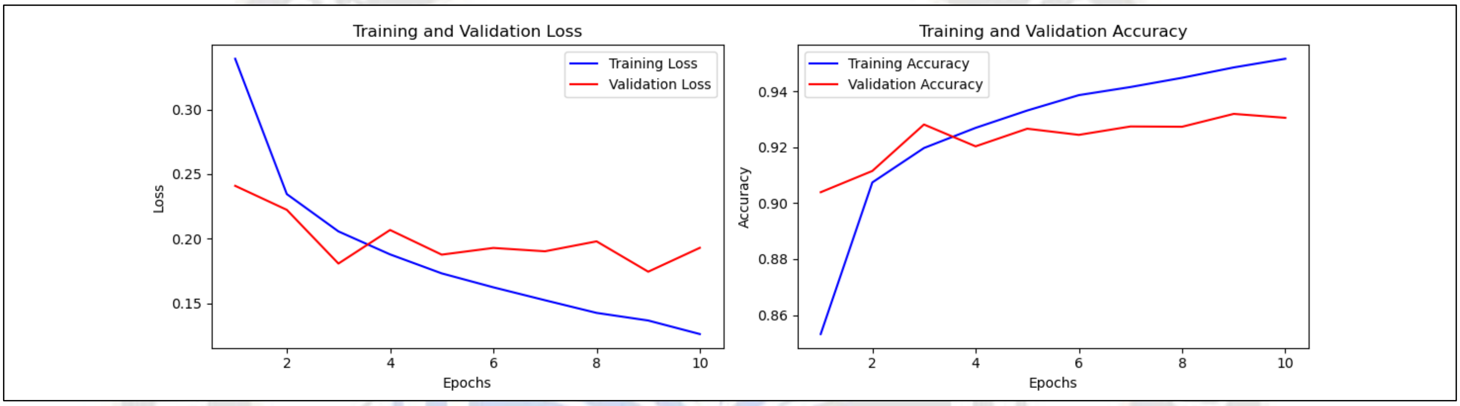

Figure 10. Model Training Metrics

To perform a comparative analysis of the CNN model with contemporary models, the Support Vector Machine(SVM) and Random Forest(RF) models were also trained on the CIFAKE dataset. Fig. 11 below presents a bar plot comparing the accuracies of the said models. The CNN model achieved an accuracy of 93.55%, surpassing the accuracies of the SVM and RF models which were 84.84% and 83.41% respectively. This demonstrates the superior performance of the CNN model in distinguishing between AI-generated and real-world images.

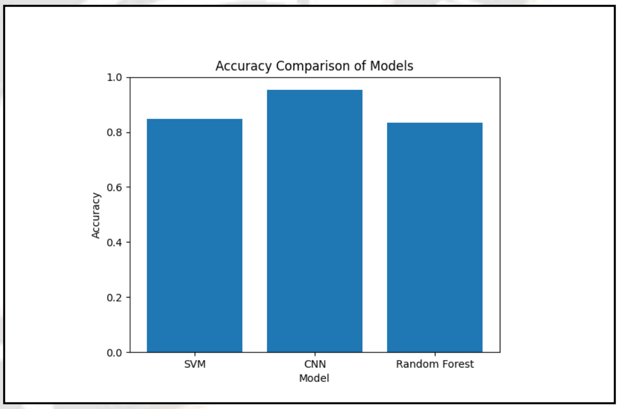

Figure 11. Accuracy Comparison Plot

The confusion matrix for each of the models is shown in the labelled Fig. 12, Fig. 13 and Fig. 14 and a tabular summary for performance comparison of those confusion matrices is shown in the table I below. Here, AI stands for Positive while Real stands for Negative in the conventional machine learning terminology.

TABLE I. PERFORMACE COMPARISON

Model	True AI	True Real	False AI	False Real
CNN	9498	9220	780	502
SVM	8545	8422	1578	1455
RF	8723	7959	2041	1277

Based on these results, we can conclude that the CNN model outperforms both the SVM and Random Forest models in terms of accurately classifying AI-generated and real images. It achieved the highest counts of true predictions(18718) and the lowest counts of false predictions(1292), indicating superior performance.

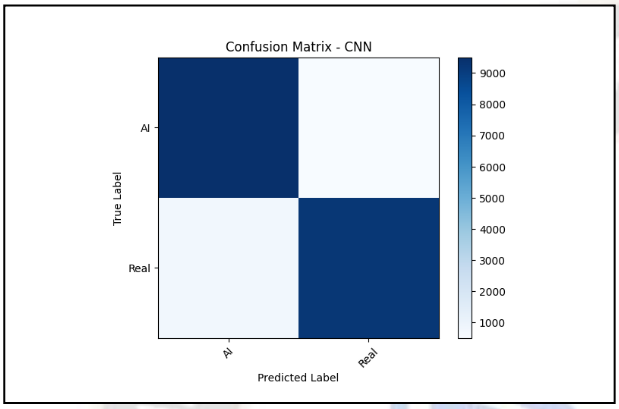

Figure 12. Confusion Matrix (CNN Model)

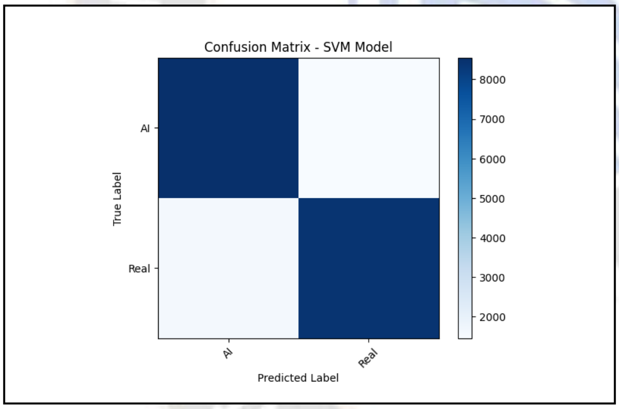

Figure 13. Confusion Matrix (SVM Model)

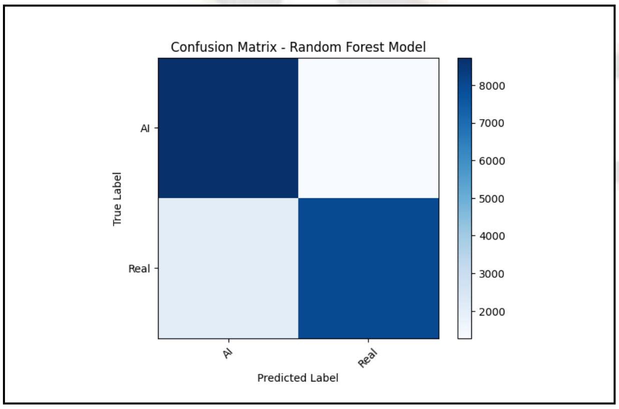

Figure 14. Confusion Matrix (SVM Model)

The receiver operating characteristic (ROC) curve for the models is shown in Fig. 15. The curve illustrates the trade-off between the true positive rate and the false positive rate. The area under the curve (AUC) value provides a measure of the models' overall performance, with a higher value indicating better discrimination between classes. In our case, the CNN model achieved an AUC of 0.9359 while the SVM and Random Forest models achieved an AUC of 0.9267 and 0.9132.

The Precison-Recall curve comparing the three models is shown in Fig. 16. Precision, which evaluates the fraction of accurately predicted AI-generated images in comparison to all predicted AI-generated images, was 0.9469 for the CNN model. This high precision score indicates that the model has a low rate of false positives, which reduces the possibility of misclassifying actual images as AI-generated. Furthermore, the model has a recall of 0.9122, which represents the proportion of properly predicted AI-generated images out of all actual AI-generated images. This high recall number implies that the model has a low false negative rate, effectively detecting the vast majority of AI-generated images in the dataset.

Lastly, Fig. 17 presents the comparative F1 score plot. The F1 score combines precision and recall into a single metric, providing an overall measure of each of the model's performance.

The proposed CNN model thus demonstrated remarkable performance in detecting AI-generated images on the CIFAKE dataset. It outperformed the SVM and RF models in terms of accuracy, precision, recall, AUC, and F1 score.

V. CONCLUSION

The rise in use of AI generated content has accelerated the pace and volume of generated information available on the web. This makes it important to detect which data is human developed and which is not. AI generated text detection follows a different approach by using a combination of large language models, style analysis and visual analysis to detect the text generated by AI models themselves. AI generated images can be recognised by analysing the pixels. Unusual colors or patterns and aberrations also provide clues for content to be synthetic. Deepfake content can be identified in one way by taking into consideration the biological signals produced naturally which can be overlooked while developing the synthetic data. Eye gazes, blood palpitations and generic human expressions are parameters which can be distinguished. Owing to all the research presented in this paper it can be concluded that AI generated content is becoming the future and currently it is very difficult to develop a model that can identify this content with 100 percent accuracy. Many solutions will be developed such as watermarking technologies and cryptographically signing videos and images.

Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 September 2023*

However, the systems that generate synthetic content will also become more robust. So, validating the content before sending and responsible use are the measures to be taken as everyone will have to adapt to the world with co-existence of both kinds of content.

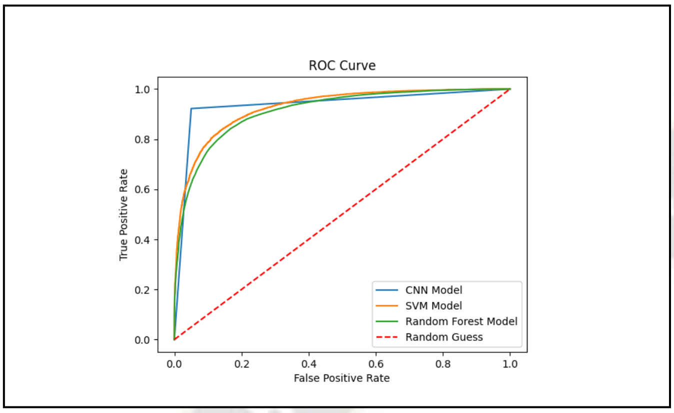

Figure 15. ROC Curve

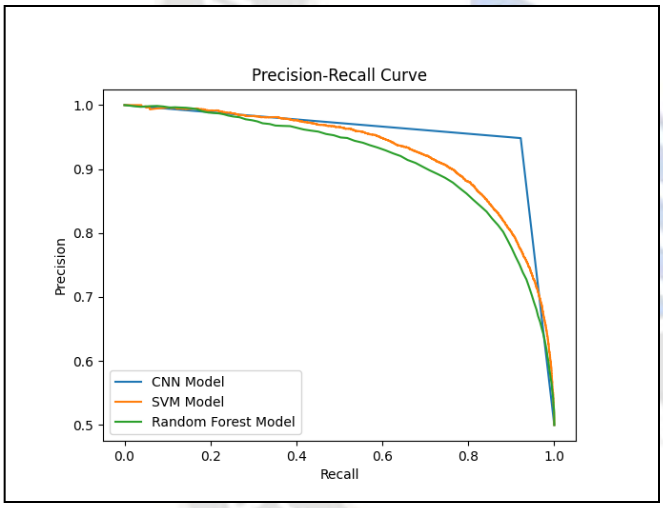

Figure 16. Precision Recall Curve

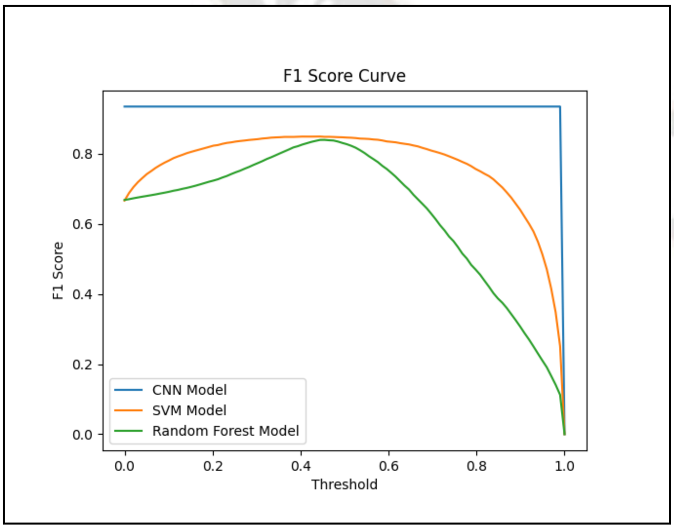

Figure 17. F1 Score Plot

VI. FUTURE SCOPE

The field of Generative AI is progressing rapidly and there are many directions in which this study can be extended. One direction is working with larger and varied datasets and examining the scalability and architecture of these models since each of these model's success is strongly reliant on the quality and diversity of the datasets they are trained on. Another direction is the addition of the audio modality and subsequently videos or deepfakes which we have briefly mentioned about along with fusing all of these models together for better and robust detection by investigating advanced techniques. Furthermore, developing a real-time detection framework can be a valuable future direction. Finally, while the models make accurate predictions, their complicated architecture may make them difficult to interpret. Using interpretability strategies, such as attention mechanisms or visualizing the learned characteristics, will help improve the explainability of the model's decisions.

REFERENCES

[1] M. S. Rana, M. N. Nobi, B. Murali and A. H. Sung, "Deepfake Detection: A Systematic Literature Review," in IEEE Access, vol. 10, pp. 25494-25513, 2022, doi:10.1109/ACCESS.2022.3154404.

[2] Coccomini, Davide Alessandro, et al. "Combining efficientnet and vision transformers for video deepfake detection." Image Analysis and Processing–ICIAP 2022: 21st International Conference, Lecce, Italy, May 23–27, 2022, Proceedings, Part III. Cham: Springer International Publishing, 2022.

[3] Deepfake Detection Challenge | Kaggle Contest https://www.kaggle.com/competitions/deepfake-detection-challenge/discussion/145721

[4] Bonettini, Nicolo, et al. "Video face manipulation detection through ensemble of cnns." 2020 25th international conference on pattern recognition (ICPR). IEEE, 2021.

[5] Mathews, S., Trivedi, S., House, A. et al. An explainable deepfake detection framework on a novel unconstrained dataset. Complex Intell. Syst. (2023).

[6] Dagar, D., Vishwakarma, D.K. A literature review and perspectives in deepfakes: generation, detection, and applications. Int J Multimed Info Retr 11, 219–289 (2022).

[7] Ma, Yongqiang, Jiawei Liu, and Fan Yi. "Is this abstract generated by ai? a research for the gap between ai-generated scientific text and human-written scientific text." arXiv preprint arXiv:2301.10416 (2023).

[8] Bonettini, Nicolo, et al. "Video face manipulation detection through ensemble of cnns." 2020 25th international conference on pattern recognition (ICPR). IEEE, 2021.

[9] How to spot AI-generated text | MIT Technology Review https://www.technologyreview.com/2022/12/19/1065596/how-to-spot-ai-generated-text/

[10] New AI classifier for indicating AI-written text | OpenAI https://openai.com/blog/new-ai-classifier-for-indicating-ai-written-text

[11] Yongqiang Ma, Jiawei Liu, Fan Yi, Qikai Cheng, Yong Huang, Wei Lu, Xiaozhong Liu (2023), "AI vs. Human -- Differentiation Analysis of Scientific Content Generation", arXiv:2301.10416

[12] ML Olympiad - Detect ChatGPT Answers | Kaggle Contest https://www.kaggle.com/competitions/ml-olympiad-detect-chatgpt-answers/data

[13] Vinu Sankar Sadasivan and Aounon Kumar and Sriram Balasubramanian and Wenxiao Wang and Soheil Feizi (2023), "Can AI-Generated Text be Reliably Detected?", arXiv:2303.11156

[14] John Kirchenbauer and Jonas Geiping and Yuxin Wen and Jonathan Katz and Ian Miers and Tom Goldstein (2023), "A Watermark for Large Language Models", arXiv:2301.10226

[15] Chandrasegaran, Keshigeyan, Ngoc-Trung Tran, and Ngai-Man Cheung. "A closer look at fourier spectrum discrepancies for cnn-generated images detection." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2021.

[16] Bird, Jordan J., and Ahmad Lotfi. "CIFAKE: Image Classification and Explainable Identification of AI-Generated Synthetic Images." arXiv preprint arXiv:2303.14126 (2023).

[17] The CIFAR-10 AND CIFAR-100 Dataset Source https://www.cs.toronto.edu/~kriz/cifar.html

[18] C. Anitha Mary and A. Boyed Wesley. "Optimization Assisted Autoregressive Method with Deep Convolutional Neural Network-Based Entropy Filter for Image Demosaicing." ICTACT Journal on Soft Computing, vol. 13, issue 3, pp. 2977-2985 (2023)

[19] Alexander Turchin, Stanislav Masharsky and Marinka Zitnik. "Comparison of BERT implementations for natural language processing of narrative medical documents". Informatics in Medicine Unlocked, vol. 36, 101139 (2023).

[20] Ravichandra Madanu, Maysam F. Abbod, Fu-Jung Hsiao, Wei-Ta Chen and Jiann-Shing Shieh. "Explainable AI (XAI) Applied in Machine Learning for Pain